CLAIMS

We claim:

1. A rule processor for conducting contextual searches, the processor comprising:

a plurality of input payload search registers;

search execution engine coupled to the plurality of search registers to perform one or more contextual searches on content in the search registers via parallel pattern matching in response to executing one or more search instructions specifying the one or more pattern searches and presenting one or more patterns to the content in the search registers.

2. The rule processor defined in Claim 1 wherein fields of the one or more search instructions are coupled to the plurality of search registers and the search execution engine.

3. The rule processor defined in Claim 1 wherein at least one of the one or more search instructions specifies a pattern that is to be searched against the content in the plurality of search registers and zero or more search parameters.

4. The rule processor defined in Claim 3 wherein one parameter specifies a portion of the pattern to be masked to enable a subset of the pattern to be searched against the content in the search registers.

5. The rule processor defined in Claim 4 wherein the portion of the pattern to be masked is specified by a mask vector to mask off specific bytes in the pattern.

6. The rule processor defined in Claim 4 wherein the zero or more parameters specify starting and ending locations that constitute a range of the content within the search registers within which the search execution engine is to constrain a search.

7. The rule processor defined in Claim 4 wherein the at least one of the instructions specifies a windowed-find-first-forward search.

8. The rule processor defined in Claim 4 wherein the at least one of the instructions specifies a windowed-find-first-reverse search.

9. The rule processor defined in Claim 1 wherein the search execution engine generates at least one result output indicative of success in searching the content in the search registers.

10. The rule processor defined in Claim 4 wherein the at least one result output comprises an indication of whether or not a match occurred between a pattern specified in at least one of the instructions and the content in the plurality of search registers.

11. The rule processor defined in Claim 4 wherein the at least one result output comprises an indication of a location in the plurality of search registers where a match occurred between a pattern specified in at least one of the instructions and the content in the search registers.

12. The rule processor defined in Claim 1 wherein at least one search instruction includes a field that specifies a parameter to use to control the search or a pointer into a memory that stores the parameter to control the search.

13. The rule processor defined in Claim 12 wherein the pointer points to a general purpose register.

14. The rule processor defined in Claim 12 wherein the value to which the pointer points is a result of a previously performed search by the search execution hardware.

15. The rule processor defined in Claim 12 wherein the value corresponds to one of a group that includes a mask, a search window parameter, and a control parameter.

16. The rule processor defined in Claim 1 wherein the plurality of input payload search registers comprises a register file.

17. The rule processor defined in Claim 16 wherein the register file comprises 2K entries of one byte each.

18. The rule processor defined in Claim 16 wherein the register file comprises a plurality of entries addressed by 11-bit register addresses.

19. The rule processor defined in Claim 1 further comprising a memory to store one or more search instructions to be applied to data in the search registers.

20. The rule processor defined in Claim 1 wherein the search instructions cause the search execution engine to perform searches for arbitrarily long patterns in the content in the search registers.

21. The rule processor defined in Claim 1 further comprising an instruction sequencer for applying one or more search instructions to the search execution engine.

22. The rule processor defined in Claim 21 wherein the one or more search instructions specify at least one pattern, range control, and program control flow.

23. The rule processor defined in Claim 21 wherein the one or more search instructions include a pointer to specify a memory location that stores information that specifies at least one pattern, range control, and program control flow.

24. The rule processor defined in Claim 21 wherein at least one search instruction in the one or more search instructions comprises opcode information to indicate a search operation type, pattern information to specify a pattern to be located, a mask to specify a portion of the pattern information that comprises the pattern, and a pair of offsets to specify starting and ending bounds of locations in the search registers for the search for the at least one search instruction.

25. The rule processor defined in Claim 1 wherein the search execution engine comprises a first output indication indicative of search success of execution of one search

instruction and a second output indication indicative of a location within the search registers of a pattern specified by the one search instruction.

26. The rule processor defined in Claim 1 wherein the search execution engine comprises:

a search array coupled to the plurality of input payload search registers, wherein content in the plurality of search registers is replicated and stored in the search array; and

a sorter coupled to the search array to perform the one or more operations in response to information specified by one or more search instructions.

27. The rule processor defined in Claim 26 wherein the search array comprises M match lines with each of the M match lines associated with a group of data stored in the search array and being indicative of whether a pattern specified by one of the one or more search instructions matches data in its associated group of data stored in the search array.

28. The rule processor defined in Claim 27 wherein the sorter is coupled to receive the M match lines to perform the one or more operations associated with matches indicated by the M match lines.

29. The rule processor defined in Claim 28 wherein the information specifies a range, and the sorter sorts the M match lines only in the specified range.

30. The rule processor defined in Claim 29 wherein the range is specified in the search instruction.

31. The rule processor defined in Claim 29 wherein the information specifies a location in a memory at which the range is stored.

32. The rule processor defined in Claim 31 wherein the memory is a register file.

33. The rule processor defined in Claim 26 wherein the search array comprises a first input to receive bits including the pattern and a second input for a mask, the search array to mask zero or more bits of the bits corresponding to the pattern based on the mask.

34. The rule processor defined in Claim 33 wherein the bits including the pattern comprises N bytes and the mask comprises N bits, each of the N bits being associated with a different one of the N bytes, wherein the search array masks one of the N bytes of the pattern when its associated bit of the N bit mask is in a first state.

35. The rule processor defined in Claim 26 wherein the sorter has a first output indicating whether one or more of the match lines match and a second output indicative of a result of performing the one or more operations.

36. The rule processor defined in Claim 35 wherein the second output is indicative of a location into the search array of a first occurrence of a match between the pattern and data stored in the search array in relation to one side of the search array.

37. The rule processor defined in Claim 35 wherein the second output is indicative of a number of matches in a range of the M match lines.

38. The rule processor defined in Claim 26 wherein the sorter operates in response to a range mask to mask out one or more outputs of the search array for a search operation.

39. The rule processor defined in Claim 26 wherein the sorter further comprises:

a priority encoder to identify a location in the search array corresponding to the M match lines corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to one side of the search array.

40. The rule processor defined in Claim 39 wherein the priority encoder is an ascending priority encoder and the one side of the search array is the top of the search array.

41. The rule processor defined in Claim 39 wherein the priority encoder is a descending priority encoder and the one side of the search array is the bottom of the search array.

42. The rule processor defined in Claim 26 wherein the sorter further comprises a counter to determine a number of matches in the search array.

43. The rule processor defined in Claim 26 wherein the sorter further comprises:

an ascending priority encoder to identify a location in the search array corresponding to the M match lines corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a top side of the search array;

a descending priority encoder to identify a location in the search array corresponding to the M match lines corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a bottom side of the search array;

a counter to determine a number of matches in a range of the M match lines; and

a selector coupled to the ascending priority encoder, the descending priority encoder and the counter and having a first output, the selector operable to select an output of the ascending priority encoder, the descending priority encoder and the counter as the first output of the sorter.

44. The rule processor defined in Claim 43 wherein the selector has a second output indicating if a match occurred between the pattern and data in the search array.

45. The rule processor defined in Claim 26 wherein the search array comprises:

a plurality of rows of memory locations to store bytes of data;

a plurality of rows of byte comparators to compare bytes of the data stored in the plurality of rows of memory locations with bytes of the pattern, each comparator of the plurality of rows of byte comparators having an output;

a plurality of masked reduction units, each of the plurality of masked reduction units coupled to receive byte masks and comparator outputs of comparators in one row of byte comparators, the plurality of masked reduction units masking individual comparator outputs based on the byte masks and combining unmasked comparator outputs for each row into one of a plurality of mask lines.

46. The rule processor defined in Claim 1 further comprising:

a rule memory to store a plurality of rules;

a rule sequencer coupled to the rule memory to select one or more rules for execution; and

a decoder to decode the one or more rules selected by the rule sequencer, the decoder coupled to the search array and sorter to provide decoded information to the search array and the sorter.

47. A rule engine content processor comprising:

a search array to perform pattern matching between data stored in the search array and an N byte pattern from the search instruction received on a first input, the search array having M match lines as outputs with each of the M match lines associated with a group of data stored in the array and being indicative of whether the N byte pattern matches data stored in its associated group of data stored in the search array; and

a sorter coupled to receive the M match lines to perform one or more operations associated with matches indicated by the M match lines, the one or more operations being performed in response to information specified by the rule, and further wherein the sorter outputs data indicative of any match found.

48. The rule engine content processor defined in claim 47, wherein the sorter has a first output indicating whether one or more of the match lines match and a second output indicative of a result of performing the one or more operations.

49. The rule engine content processor defined in claim 47 wherein the search array includes a second input to receive an N bit mask to apply to the N byte pattern.

50. The rule engine content processor defined in Claim 47 wherein the sorter comprises a range mask to mask out a portion of the M mask lines based on a range.

51. The rule engine content processor defined in Claim 50 wherein the range mask logically ANDs the M mask lines with a pair of offsets specified by the search instruction.

52. The rule engine content processor defined in Claim 51 wherein the rule includes the pair of offsets.

53. The rule engine content processor defined in Claim 50 wherein the rule includes a pointer to a location in a memory where the offsets are stored.

54. The rule engine content processor defined in claim 47 wherein the data output for the sorter is feedback for use and an input to the sorter in the next cycle.

55. The rule engine content processor defined in Claim 47 wherein the sorter further comprises:

a priority encoder to identify a location in the search array corresponding to the M match lines corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to one side of the search array.

56. The rule engine content processor defined in Claim 55 wherein the priority encoder is an ascending priority encoder and the one side of the search array is the top of the search array.

57. The rule engine content processor defined in Claim 55 wherein the priority encoder is a descending priority encoder and the one side of the search array is the bottom of the search array.

58. The rule engine content processor defined in Claim 47 wherein the sorter further comprises a counter to determine a number of matches in a range of the M match lines.

59. The rule engine content processor defined in Claim 47 wherein the sorter further comprises:

an ascending priority encoder to identify a location in the search array corresponding to the M match lines corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a top side of the search array;

a descending priority encoder to identify a location in the search array corresponding to the M match lines corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a bottom side of the search array;

a counter to determine a number of matches in a range of the M match lines; and

a selector coupled to the ascending priority encoder, the descending priority encoder and the counter and having a first output, the selector operable to select an output of the ascending priority encoder, the descending priority encoder and the counter as the first output of the sorter.

60. The rule engine content processor defined in Claim 55 wherein the selector has a second output indicating if a match occurred between the pattern and data in the search array.

61. The rule engine content processor defined in Claim 47 wherein each of the N bits is associated with a different one of the N bytes, wherein the search array masks one of the N bytes of the pattern when its associated bit of the N bit mask is in a first state.

62. The rule engine content processor defined in Claim 47 wherein the information specifies a range, and the sorter sorts the M match lines only in the specified range.

63. The rule engine content processor defined in Claim 62 wherein the range is specified in the search instruction.

64. The rule engine content processor defined in Claim 62 wherein the information specifies a location in a memory at which the range is stored.

65. The rule engine content processor defined in Claim 47 wherein the search array comprises:

a plurality of rows of memory locations to store bytes of the data;

a plurality of rows of byte comparators to compare bytes of the data stored in the plurality of rows of memory locations with bytes of pattern, each comparator of the plurality of rows of byte comparators having an output;

a plurality of masked reduction units, each of the plurality of masked reduction units coupled to receive byte masks and comparator outputs of comparators in one row of byte comparators, the plurality of masked reduction units masking individual comparator outputs based on the byte masks and combining unmasked comparator outputs for each row into one of the M mask lines.

66. The rule engine content processor defined in Claim 47 further comprising:

a rule memory to store a plurality of search instructions;

a rule sequencer coupled to the search instruction memory to select one or more search instructions for execution; and

a decoder to decode the one or more search instructions selected by the rule sequencer, the decoder coupled to the search array and sorter to provide decoded information to the search array and the sorter.

67. A process comprising:

loading a set of input payload search registers with content;

presenting a pattern identified by a search instruction to be searched in the search registers;

performing pattern matching between the pattern and the content stored in the search registers; and

outputting an indication of a result of performing the pattern matching.

68. The process defined in Claim 67 further comprising:

generating a plurality of match lines associated with rows of the search array, wherein match lines of the plurality of match lines indicating whether a match occurred between the pattern and data in a row associated with one of the match lines;

performing the one or more operations on at least a set of the match lines in response to information specified by the search instruction;

outputting an indication as to whether one or more of the match lines match the pattern and a result of performing the one or more operations.

69. The process defined in Claim 67 wherein loading the search registers is performed to store, replicate, and interleave data such that data for one row is stored in an adjacent row in shifted form.

70. The process defined in Claim 67 further comprising:

converting code into a sequence of search instructions;

executing the sequences of search instructions in consecutive cycles such that pattern matching is performed for each of the plurality of search instructions.

71. A process for performing contextual searches in a pipelined fashion, the process comprising:

fetching a rule from a rule memory;

decoding the rule and assembling indirect fields, if any;

executing one or more search operations on values in a plurality of input payload search registers storing content; and

performing sort operations on results of executing the one or more search operations.

72. The process defined in Claim 71 wherein the process is performed in a four stage pipeline with a search array and a sorter.